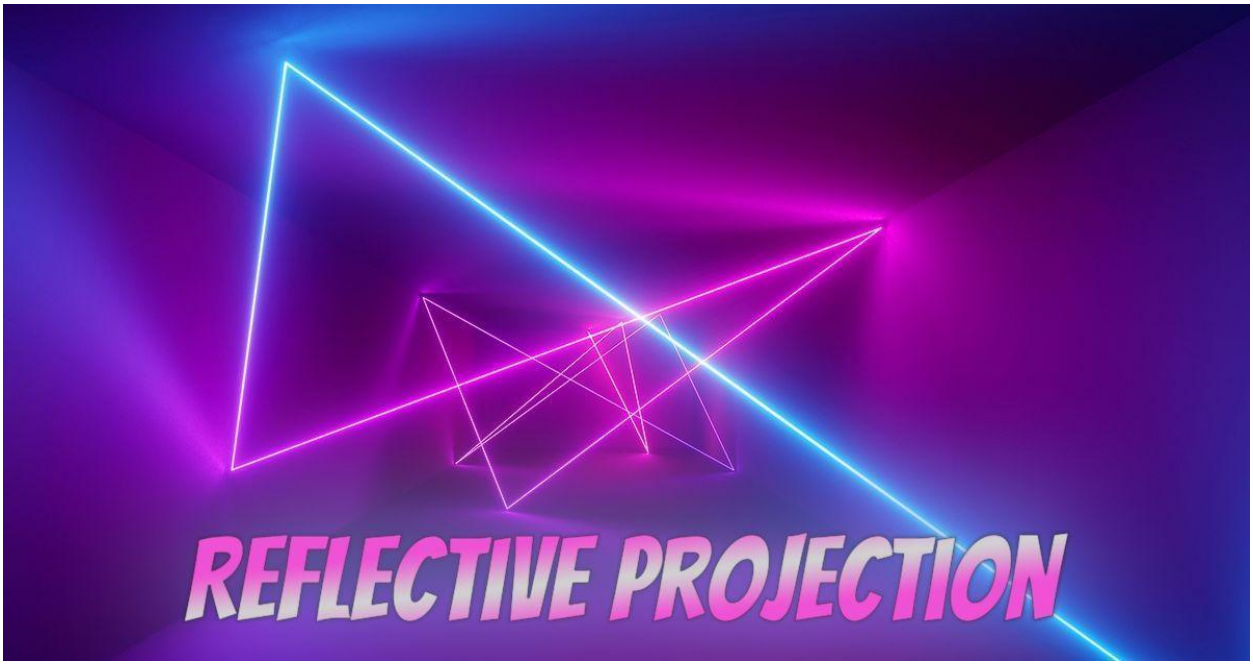

Setup and Information Guide for Reflective Projection

For support & questions:

Visit the Discord server here
Or email me directly at: pathiralgames@gmail.com

GETTING STARTED:

1. Make a new **cube** game object and give it a big scale. Make sure a box collider is in place.
2. Select the cube and make sure it's layer is set on whatever the reflective layer should be. For now you can leave it to Default.
3. We're done making a reflective object.
4. Drag the *Parent Emitter* prefab and place it in the scene. Make sure the cube is in front of it so the projection hits the cube.
5. Select the child *emitter* inside the parent object. You will find the **Reflective Projection** script. In here you can add the amount of *reflections* needed. Set it to 5.
6. You can also set the *Max Distance* the projection travels before it hits a reflective object. Set it to 50 (make sure 50 is enough that it hits the cube).
7. The *Reflective Layer* is the layer that the reflective objects should have, indicating that they are reflective objects to begin with. Set it to 0. Since we set our object to the Default layer and Default is 0.
8. You can make your own *Projection Material* and set it in that field to give the projection the color and/or effects of your choice. Just leave it empty. When the field is empty it automatically draws back to the default material found in Resources that comes with the package.
9. Play the game.
10. Now the ray should travel at the cube and reflect off of it.
11. You have successfully created a reflective projection.

PROPERTIES & METHODS:

reflectionPoints (int) -> returns the number of current reflections.

ricochetFinished (*Bool*) -> return whether the ricochet has finished or not.

GetHitObjects() -> returns a list of type GameObject with all the game objects of reflective layer it hit.

GetHitStopObjects() -> returns a list of type GameObject of all the game objects with stop reflection layer hit.

OBJECTS DETECTING PROJECTION HIT:

The script got you covered. It comes with a public property called **GetHitObjects** which returns a list containing all the objects hit.

1. Create a script and attach it to the objects you want to detect the projection when hit.
2. Get the *Reflective Projection* script and store it in a variable called script.
3. In the Update loop do:

```
if (script.GetHitObjects.Contains(gameObject)){
        //do whatever
        //this will trigger everytime a projection hits/ricochets on an object
        //with this script
}
```

And this is how you can easily detect whether a projection has hit your certain/group of object(s)

Same thing with GetHitStopObjects() to find the objects hit with the Stop Reflection Layer.